REMARKS

This Application has been carefully reviewed in light of the Final Office Action dated January 28, 2008 (the "*Final Office Action*"). At the time of the *Final Office Action*, Claims 2-12 and 15-45 were pending and rejected in the Application. As described below, Applicant believes all claims to be allowable over the cited references. Therefore, for the reasons discussed below, Applicant respectfully requests reconsideration and full allowance of all pending claims.

Finality of Office Action

Applicant respectfully requests withdrawal of the finality of the *Final Office Action* delivered January 28, 2008. The M.P.E.P provides guidelines for when a final rejection is proper on a second action:

> Under present practice, second or any subsequent actions on the merits shall be final, except where the examiner introduces a new ground of rejection that is neither necessitated by Applicant's amendment of the claims nor based on information submitted in an information disclosure statement filed during the period set forth in 37 CFR 1.97(c) with the fee set forth in 37 CFR 1.17(p).

M.P.E.P. § 706.07(a) (emphasis added). In the present case, Applicant contends that within the *Final Office Action* dated January 28, 2008, the Examiner has introduced new grounds of rejection that were not necessitated by Applicant's amendment of the claims.

Specifically, and with regard to independent Claim 45, Applicant notes that in the previous Office Action mailed August 7, 2007, the Examiner rejected Claim 45 over U.S. Patent No. 7,146,544 issued to Hsu et al. ("*Hsu*") in view of U.S. Patent Application Publication No. 2005/0015472 issued to Catania et al. ("*Catania*"). In the subsequent Response to Office Action submitted by Applicant on October 26, 2007, Applicant argued the patentability of independent Claim 45 over the proposed *Hsu-Catania* combination but did not amend independent Claim 45. In the *Final Office Action*, the Examiner rejects the claim over *Hsu* in view of U.S. Patent Application Publication No. 2005/0172306 issued to Agarwal et al. ("*Agarwal*") and *Catania*. Thus, the proposed *Hsu-Agarwal-Catania* combination, as applied to Claim 45, is a new ground of rejection. The Examiner states that

"Applicant's amendment necessitated the new ground(s) of rejection presented in this Office Action." (*Final Office Action*, page 14). However, since Applicant's Claim 45 was not amended in the Response to Office Action submitted on October 26, 2007, the new grounds of rejection provided by the Examiner in the *Final Office Action* could not have been necessitated by Applicant's amendment to the claims. As such, Applicant contends that the Examiner has introduced a new ground of rejection to Claim 45, which was not necessitated by Applicant's amendment of the claim. Applicant respectfully submits that the finality of the *Final Office Action* is improper.

Additionally, Applicant notes that in the Response to Office Action submitted on October 26, 2007, Applicant only amended Claim 2 by rewriting Claim 2 in independent form. Specifically, the claim elements recited in independent Claim 1, as originally filed, were added to Claim 2. The original claim elements of Claim 2 were otherwise unaltered. Thus, the amendment to Claim 2 did not change the scope of Claim 2. Stated differently, the amendment to Claim 2 did not result in the recitation of additional subject matter that would require a new search by the Examiner. Accordingly, the new rejection of Claim 2 over the proposed *Hsu-Agarwal* combination constitutes a new ground of rejection that was not necessitated by Applicant's amendment of the claim. Applicant respectfully submits that the finality of the *Final Office Action* is improper.

For at least these reasons, Applicant respectfully requests that the finality of the *Final Office Action* dated January 28, 2008, be withdrawn.

Section 103 Rejections

The Examiner rejects Claims 2, 3-15, 16-26, and 29-44 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,146,544 issued to Hsu et al. ("*Hsu*") in view of U.S. Patent Application Publication No. 2005/0172306 issued to Agarwal et al. ("*Agarwal*"). The Examiner rejects Claims 27, 28, and 45 under 35 U.S.C. § 103(a) as being unpatentable over *Hsu* in view of *Agarwal*, and in further view of U.S. Patent Application Publication No. 2005/0015472 issued to Catania et al. ("*Catania*"). Applicant notes that Claims 13 and 14 were cancelled in the previous Response to Office Action submitted on October 26, 2007.

For the reasons discussed below, Applicant respectfully requests reconsideration and allowance of Claims 2-12 and 15-45.

A. Claims 2-12, 15-28, and 29-44

Independent Claim 2 has not been amended and recites:

> A method for managing faults in a web service architecture comprising:
> receiving a service request in a web service language, wherein the service request comprises invoking a service over a network;
> *translating the service request into a non-web service language*;
> executing the service request;
> encountering an exception during the execution, wherein the execution comprises a fault preventing the fulfillment of the service request;
> persisting the fault; and
> providing a fault response.

Applicant respectfully submits that the proposed *Hsu-Agarwal* combination does not disclose, teach, or suggest the combination of claim elements recited in Claim 2, as previously presented.

For example, the proposed *Hsu-Agarwal* combination does not disclose, teach, or suggest "translating the service request into a non-web service language," as recited in Claim 2. In the *Office Action*, the Examiner acknowledges that *Hsu* fails to disclose the recited claim features and instead relies on *Agarwal*. Specifically, the Examiner cites paragraphs 1, 42, and 65-69 of *Agarwal* and states that *"Agarwal* discloses the additionally recited feature of translating the service request into a non-web service language (i.e., Structured Query Language {SQL} [0042][0065-0069]." (*Final Office Action*, page 4). Applicant respectfully disagrees, however, with the Examiner's finding that *Agarwal* discloses the recited claim language.

Agarwal merely discloses that "[m]any service components have dependencies on other service components - such that failures occurring in one service component affect other services and ultimately customer applications." (*Agarwal*, page 1, paragraph 2). As

disclosed in *Agarwal*, "[t]he identification and tracking of dependencies between components of distributed systems is becoming increasingly important for integrated fault management (problem determination, impact analysis and repair for a set of cooperating components or processes)." (*Agarwal*, page 1, paragraph 2). Accordingly, *Agarwal* relates to "methods, apparatus, and computer programs for determining run-time dependencies between logical components of a data processing environment." (*Agarwal*, Abstract). Specifically, the methods described in *Agarwal* "may be used to identify runtime dependencies between, for example, servlets, Web server processes serving Uniform Resource Locators (URLs), Enterprise JavaBeans (EJBs), and Structure Query Language (SQL) server processes associated with a Web application." (*Agarwal*, page 3, paragraph 42). More specifically, the methods described in *Agarwal* may be used to identify "servlet-to-EJB, servlet-to-SQL, and EJB-to-SQL dependencies." (*Agarwal*, page 5, paragraph 65). Thus, each of a servlet, an EJB, and a SQL request is a component, and *Agarwal* relates to identifying component-to-component (i.e., servlet-to-EJB, servlet-to-SQL, and EJB-to-SQL) dependencies. *Agarwal* does not relate to "translating the service request into a non-web service language," as recited in Claim 2.

The portions of *Agarwal* cited by the *Final Office Action* merely clarify the methods discussed above. Specifically, *Agarwal* discloses a "monitoring infrastructure 130 and API 132 [that] provide a framework to extract monitoring data from the Web application server (WAS) 100 and local application components 120 at runtime." (*Agarwal*, page 5, paragraph 68). As disclosed in *Agarwal*, "the monitoring infrastructure 130 on the server-side . . . keeps the performance data as raw counter values" and "the agents 20, 22 on the client side . . . retrieve and manipulate the raw counters to provide more meaningful statistics, such as average, time-weighted average, percentage change, rate, etc." (*Agarwal*, page 5, paragraph 68). Thus, the client side generates statistics from raw data. (*Agarwal*, page 5, paragraph 68). For example, *Agarwal* states:

> In the present embodiment, the agent 20 obtains activity metrics including access counts and average response times of servlets and EJBs, and obtains accessed URLs from an HTTP access log. The dependence of a URL on the servlet which serves that URL is obtained from a Web application's configuration files. The agent 20 provides to the correlation

> identifier events for URLs, servlets, and EJBs; whereas agent 22 provides events for SQL request executions.

(*Agarwal*, page 5, paragraph 69). As such, *Agarwal* merely describes that an agent on the client side extracts data such as access counts, response times, accessed URLs, HTTP access logs and configuration files to identify servlet-to-EJB, servlet-to-SQL, and EJB-to-SQL dependencies. Identifying servlet-to-EJB, servlet-to-SQL, and EJB-to-SQL dependencies, however, is not analogous to Applicant's step of "translating the service request into a non-web service language," as recited in Claim 2.

In fact, Applicant submits that there is no disclosure in *Agarwal* of "translating" anything. As cited by the Examiner, *Agarwal* discloses that "[a] Web application may issue SQL queries to a database server." (*Agarwal*, page 2, paragraph 25). However, there is no disclosure in *Agarwal* that the SQL queries are translated to/or from any other format. To the extent that the Examiner is identifying the "SQL queries" disclosed in *Agarwal* as being analogous to Applicant's "non-web service language" (which Applicant does not admit or deny), there is no disclosure in *Agarwal* that such SQL requests originate as something else and then are translated into SQL language. The converse is also true. There is no disclosure in *Agarwal* that the SQL queries originate as SQL requests and are then translated into another language. Accordingly, Applicant respectfully submits that there is no disclosure in *Agarwal* or in the *Hsu-Agarwal* combination of "translating the service request into a non-web service language," as recited in Claim 2.

For at least these reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 2, together with 3-10 that depend on Claim 2.

In the *Final Office Action*, the Examiner also rejects Claims 11 and 26 over the proposed *Hsu-Agarwal* combination. However, independent Claims 11 and 26 include certain features that are analogous to those discussed above with regard to Claim 2. For example, Claim 11 recites "a service interface operable to . . . translate the service request into a non-web service language." As another example, Claim 26 recites a "web service module operable to . . . translate the service request into a non-web service language." Accordingly, for reasons similar to those discussed above with regard to Claim 2, Applicant

respectfully submits that Applicant's Claims 11 and 26 are allowable over the proposed *Hsu-Agarwal* combination.

For at least these reasons, Applicant respectfully requests reconsideration and allowance of independent Claims 11 and 26, together with Claims 12 and 15-25 that depend on Claim 11 and Claims 27-45 that depend on Claim 26.

B. Independent Claim 45

Independent Claim 45 is rejected over the proposed *Hsu-Agarwal-Catania* combination. Independent Claim 45 has not been amended and recites:

> A system for managing faults in a web services architecture comprising:
> a system interface operable to receive a service request in a web services format, *the system interface further operable to translate the service request into a non-web service format*;
> a service implementation operable to fulfill the service request, generate a fault report, and persist the fault, the persistence comprising storing the fault report in a persistent store, wherein generating a fault report comprises detecting a fault during the fulfillment of the service request, and *persisting the fault comprises attaching a unique identifier to the fault report*;
> a fault service implementation operable to retrieve the fault report from the persistent store and *translate the fault report into a web service format*; and
> a fault service interface operable to receive fault service requests and transmit a fault service response.

In the previous Response to Office Action submitted on October 26, 2007, Applicant made three separate and distinct arguments as to why each of the above-emphasized claim elements were not disclosed in either *Hsu* or *Catania*. In the *Final Office Action*, the Examiner relies upon *Hsu-Agarwal-Catania* combination as a new grounds of rejection. However, the Examiner continues to apply *Hsu* and *Catania* to the claim elements emphasized above in the same manner as applied in the previous *Office Action* delivered on August 7, 2007. Because Applicant believes the previous arguments continue to have merit, Applicant reiterates and expands upon them now.

1. *The Hsu-Agarwal-Catania combination does not disclose, teach, or suggest a system interface operable to "translate the service request into a non-web service format"*

In the previous Response to Office Action, Applicant argued that the *Hsu-Catania* combination does not disclose, teach, or suggest a system interface operable to "translate the service request into a non-web service format," as recited in Claim 45. In the *Final Office Action*, the Examiner continues to identify *Catania* as disclosing the recited features and operations. (*Final Office Action*, pages 12-13) However, the Examiner's continued reliance on *Catania* is puzzling to Applicant. It is confusing to Applicant primarily because it appears that the Examiner agreed with Applicant's arguments that *Catania* does not disclose similar claim language recited in independent Claim 2 (as presented in the previous Response to Office Action submitted on October 26, 2007).[1] Applicant respectfully points out that because *Catania* did not disclose the operation of "translating the service request into a non-web service language," as disclosed in Claim 2, the Examiner removed *Catania* as a reference with respect to that claim and replaced *Catania* with *Agarwal*. Because the same arguments were equally applicable to independent Claim 45, it is unclear to Applicant whether the Examiner intended to continue to rely on *Catania* for disclosure of a system interface operable to "translate the service request into a non-web service format," as recited in Claim 45, or if the Examiner intended to rely on *Agarwal* (in a similar manner as that used to reject Claim 2 in the *Final Office Action*). Regardless, Applicant contends that neither *Catania* nor *Agarwal* disclose the recited features, and Applicant discusses the deficiencies of each reference below.

First, it continues to be Applicant's position that *Catania* does not disclose, teach, or suggest "a system interface operable to "translate the service request into a non-web service format," as recited in Claim 45. Rather, *Catania* discloses "three primary roles: service

[1] In the previous Office Action delivered on August 7, 2007, the Examiner identified *Catania* as disclosing "translating the service request into a non-web service language," as recited in Claim 2. Applicant persuasively argued in the following Response to Office Action submitted on October 26, 2007, that, in fact, *Catania* did not disclose such operations. The Examiner appears to have agreed with Applicant since the Examiner found new grounds of rejection were necessary to maintain the rejection of Claim 2. Since *Catania* did not disclose the operation of "translating the service request into a non-web service language," the Examiner removed *Catania* as a reference as applied to Claim 2 and replaced *Catania* with *Agarwal*.

provider, service registry, and service requester." (*Catania*, page 1, paragraph 7). "The service provider is the entity that provides access to the Web service and **publishes the service description in a service registry**." (*Catania*, page 1, paragraph 7, emphasis added). By contrast, **"[t]he service requestor finds the service description in a service registry** or other location and can use the information in the description to bind to a service." (*Catania*, page 1, paragraph 7, emphasis added). With regard to the messages that are sent, *Catania* discloses that "[w]eb services typically send XML messages formatted in accordance with the Simple Object Access Protocol (SOAP) specification." (*Catania*, page 1, paragraphs 8). *Catania* further clarifies:

> The XML messages are described using the Web Services Description Language (WSDL) specification, which, along with the Universal Description Discovery and Integration (UDDI) registry, provides a definition of the interface to a Web service and identifies service providers in a network. The WSDL specification is an XML-based language used to define Web services and describe how to access them. **An application trying to use a particular Web Service can often use WSDL to find the location of the Web service, the function calls available, and the format that must be followed to access the Web service. Therefore, the client first obtains a copy of the WSDL file from the server and then uses the information in this file to format a SOAP request.**

(*Catania*, page 1, paragraphs 8-9, emphasis added). Thus, *Catania* merely discloses that web service requests are sent in the WSDL format. The service requestor must obtain a copy of the WSDL file from the server and then format the request in the proper SOAP request format prior to it being sent. Because the SOAP format is a web service format,[2] Applicant contends that using the WSDL registry to format the request in a SOAP format is not analogous to "translat[ing] the service request into a non-web service format." Furthermore, *Catania* explicitly describes that such formatting is done prior to the message being sent. Accordingly, there is no disclosure in *Catania* of a system interface that is operable to receive the service request and then "translate the service request into a non-web service format," as recited in Claim 45.

[2] Webopedia defines SOAP as "a lightweight XML-based messaging protocol used to encode the information in Web service request and response messages before sending them over a network." (See, www.webopedia.com, last visited 10/4/2007).

Second, *Agarwal* also does not disclose, teach, or suggest "a system interface operable to "translate the service request into a non-web service format," as recited in Claim 45. As discussed above with regard to Claim 2, *Agarwal* merely discloses "[t]he identification and tracking of dependencies between components of distributed systems" and specifically the identification of dependencies between "servlets, Web server processes serving Uniform Resource Locators (URLs), Enterprise JavaBeans (EJBs), and Structure Query Language (SQL) server processes associated with a Web application." (*Agarwal*, page 1, paragraph 2; page 5, paragraph 65). For the reasons discussed in detail above with regard to Claim 2, however, the mere identification of component-to-component (i.e., servlet-to-EJB, servlet-to-SQL, and EJB-to-SQL) dependencies is not analogous to Applicant's system interface that is operable to receive the service request and then "translate the service request into a non-web service format," as recited in Claim 45.

Again, Applicant submits that there is no disclosure in *Agarwal* of "translating" anything. As cited by the Examiner, *Agarwal* discloses that "[a] Web application may issue SQL queries to a database server." (*Agarwal*, page 2, paragraph 25). There is no disclosure in *Agarwal*, however, that the SQL queries are translated to/or from any other format. To the extent that the Examiner is identifying the "SQL queries" disclosed in *Agarwal* as being analogous to Applicant's "non-web service format" (which Applicant does not admit or deny), there is no disclosure in *Agarwal* that such SQL requests originate as something else and then are translated into SQL language. The converse is also true. There is no disclosure in *Agarwal* that the SQL queries originate as SQL requests and are then translated into another language. Accordingly, Applicant respectfully submits that there is no disclosure in *Agarwal* or in the *Hsu-Agarwal-Catania* combination of a system interface that is operable to receive the service request and then "translate the service request into a non-web service format," as recited in Claim 45.

For at least these reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 45.

2. *The Hsu-Agarwal-Catania combination does not disclose, teach, or suggest a service implementation operable to "persist the fault . . . wherein persisting the fault comprises attaching a unique identifier to the fault report"*

In the previous Response to Office Action, Applicant argued that the *Hsu-Catania* combination does not disclose, teach, or suggest a service implementation operable to "persist the fault . . . wherein persisting the fault comprises attaching a unique identifier to the fault report," as recited in Claim 45. In the *Final Office Action*, the Examiner continues to rely on *Hsu* for disclosure of the recited claim features. However, the cited portion of *Hsu* merely discloses:

> Exceptions may have associated error data, which may include error codes, stored in the error catalog 210. Some categories of exceptions may not need error data stored in the error catalog 210. For example, exceptions that are concrete subclasses (i.e. a subclass that may have instances or be instantiated rather than inherited) of BusinessException do not need error data in the error catalog, while exceptions that are concrete subclasses of SystemException and FrameworkException do need their error data stored. The error catalog 210 may be loaded based on information in a configuration file or files 212. Among that information are keys used for message display in an error page. When an exception occurs and an action forward is called, the error catalog 210 may be accessed based on the error code and used to determine which error message to display in the resulting page. In some cases, the error code catalog 210 may also be accessed based on the type of error action forward and used to determine the error message to display in the resulting page. With the error code catalog 210, error JSP pages may remain generic. The error message may be determined at runtime and, therefore, may be plugged into the framework of a JSP.

(*Hsu*, column 8, lines 36-54). Although the cited portion discusses the use of error codes, *Hsu* only indicates that the error codes are used to identify the type of message to display and further, indicates that the error codes are applied to "categories of exceptions." Portions of *Hsu* immediately preceding the portion cited by the Examiner state:

> The interface for directing the flow of an HTTP request in the event of an exception is based on the categories of exceptions. There may be three separate abstract subclasses of WPAException that all exceptions must extend. In this way, exceptions may be easily classified. Depending on which subclass of WPAException a specific exception falls into, one of

> two action forwards may be called, with the exception or an error code passed to it . . .

(*Hsu*, column 8, lines 18-20). Because *Hsu* merely discloses "error codes" generally and states that exceptions are classified into categories of exceptions, Applicant respectfully submits that *Hsu* and the proposed *Hsu-Catania* combination do not disclose, teach, or suggest "a service implementation operable to "persist the fault . . . wherein persisting the fault comprises attaching a unique identifier to the fault report," as recited in Claim 45.

For at least these additional reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 45.

3. *The Hsu-Agarwal-Catania combination does not disclose, teach, or suggest a fault service implementation operable to "translate the fault report into a web service format"*

In the previous Response to Office Action, Applicant argued that the *Hsu-Catania* combination does not disclose, teach, or suggest a fault service implementation operable to "translate the fault report into a web service format," as recited in Claim 45. In the *Final Office Action* the Examiner continues to provide no citation to any specific reference and instead merely states "e.g., WSDL." (*Final Office Action*, page 12). As such, the *Final Office Action* again leaves Applicant guessing as to which reference the Examiner is relying upon. For the same reasons provided in the previous Response to Office Action submitted on October 26, 2007, Applicant continues to contend, however, that neither *Hsu* nor *Catania* disclose the recited features and operations. Since the Examiner's rejection of the claim limitation is the same as the previous rejection (in which *Agarwal* was not cited as a reference), Applicant has not provided arguments with relating to *Agarwal.* Applicant is ready to do so, however, if it becomes appropriate.

To the extent that *Hsu* discloses reporting faults or providing a fault report, *Hsu* only discloses that "the error handler or manager 128 functions to track or chain errors occurring in series, catalog error messages based on error codes, and display error messages using an error catalog." (*Hsu*, column 7, lines 9-12). Applicant finds no discussion of web service formats or of translating fault reports into a web service format in *Hsu.* Certainly, the

tracking, cataloging, and displaying of errors is not analogous to providing a fault service implementation operable to "**translate the fault report into a web service format**," as recited in Applicant's Claim 45.

Catania does not make up for these deficiencies. As discussed above, *Catania* merely discloses that "[w]eb services typically send XML messages formatted in accordance with the Simple Object Access Protocol (SOAP) specification." (*Catania*, page 1, paragraphs 8). Thus, *Catania* merely discloses that web service requests are sent in the WSDL format. The service requestor must obtain a copy of the WSDL file from the server and then format the request in the proper SOAP request format prior to it being sent. There is no disclosure of providing a fault service implementation operable to "**translate the fault report into a web service format**," as recited in Applicant's Claim 45.

For at least these additional reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 45.

CONCLUSION

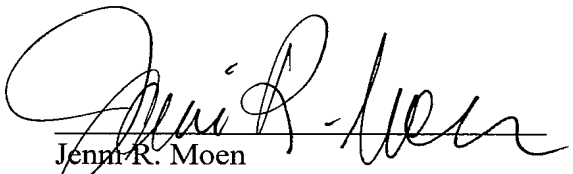
Applicant has made an earnest attempt to place this case in condition for immediate allowance. For the foregoing reasons and for other reasons clear and apparent, Applicant respectfully requests reconsideration and allowance of the pending claims.

Applicant does not believe any fees are due. However, the Commissioner is hereby authorized to charge any additional fees or credit any overpayment to Deposit Account No. 05-0765 of Electronic Data Systems Corporation.

If there are matters that can be discussed by telephone to advance prosecution of this application, Applicant invites the Examiner to contact its attorney at the number provided below.

Respectfully submitted,

Baker Botts L.L.P.
Attorneys for Applicant



Jenni R. Moen
Reg. No. 52,038
(214) 953-6809

Dated: March 11, 2008

CORRESPONDENCE ADDRESS:

at Customer No. 35005